Steering Away from Harm: An Adaptive Approach to Defending Vision Language Model Against Jailbreaks

Han Wang[1], Gang Wang[1], Huan Zhang[1]
[1]University of Illinois Urbana-Champaign
{hanw14, gangw}@illinois.edu, huan@huan-zhang.com

Abstract

Vision Language Models (VLMs) can produce unintended and harmful content when exposed to adversarial attacks, particularly because their vision capabilities create new vulnerabilities. Existing defenses, such as input preprocessing, adversarial training, and response evaluation-based methods, are often impractical for real-world deployment due to their high costs. To address this challenge, we propose ASTRA, an efficient and effective defense by adaptively steering models away from adversarial feature directions to resist VLM attacks. Our key procedures involve finding transferable steering vectors representing the direction of harmful response and applying adaptive activation steering to remove these directions at inference time. To create effective steering vectors, we randomly ablate the visual tokens from the adversarial images and identify those most strongly associated with jailbreaks. These tokens are then used to construct steering vectors. During inference, we perform the adaptive steering method that involves the projection between the steering vectors and calibrated activation, resulting in little performance drops on benign inputs while strongly avoiding harmful outputs under adversarial inputs. Extensive experiments across multiple models and baselines demonstrate our state-of-the-art performance and high efficiency in mitigating jailbreak risks. Additionally, ASTRA exhibits good transferability, defending against unseen attacks (i.e., structured-based attack, perturbation-based attack with project gradient descent variants, and text-only attack). Our code is available at https://github.com/ASTRAL-Group/ASTRA.

1. Introduction

Vision Language Models (VLMs) [8, 12, 28, 60] have attracted significant attention from both the industry and academia for their remarkable vision-language cognition capabilities [39]. Despite widespread applications, VLMs still face safety challenges due to limitations inherent in their underlying language models. Moreover, integrating visual inputs can open up a new surface for adversarial attacks. These safety issues regarding VLM have led to a lot of research on jailbreak attacks and defense strategies [17, 47, 53, 61].

Jailbreak attacks in VLMs aim to induce models to generate harmful responses by using jailbreaking image-text pairs [22–24, 26, 41, 48, 51]. These jailbreak attacks can be categorized into two types: (i) perturbation-based attacks, which create adversarial images that prompt generation of the harmful response from VLMs [2, 38, 41, 46], (ii) structured-based attacks, which embeds the malicious queries into images via typography to bypass the safety alignment of VLMs [17, 30]. Countermeasures for both attacks have been explored extensively: the input preprocessing-based method [37] or adversarial training [25] have proven effective for perturbation-based attacks. However, these defenses suffer as they require intensive computations to purify the image or fine-tune the model. Response evaluation-based [18, 53, 58] defenses have been proposed for structured-based attacks, but they all require running model inference multiple times to potentially identify harmful outputs, which dramatically increases the cost of real-world deployment.

In this work, we argue that an efficient defense framework should not require significant computational resources during training or generating responses multiple times during inference. Drawing inspiration from recent advancements in activation steering in Large Language Model (LLM) [4, 20, 43, 52], we propose ASTRA, an efficient and effective defense by adaptively steering models away from adversarial feature directions via image attribution activations to resist VLM attacks. We find that simply borrowing the method from steering LLM for safeguarding VLM is not empirically workable due to the mismatch between the steering vectors obtained from textual and visual data, which necessitates our image attribution approach.

Specifically, ASTRA consists of two steps: constructing steering vectors via image attribution, and adaptive activation steering at inference time. We seek to construct steering vectors representing the direction of harmful responses.

This can be done by constructing a set of adversarial images (e.g., using projected gradient descent (PGD) [34] algorithm) and then identifying visual tokens in each adversarial image most likely to trigger the jailbreak. To attribute such visual tokens, we fit a linear surrogate model using Lasso and estimate the impact of the inclusion/exclusion of each visual token on the probability of jailbreaks. The top-k impactful visual tokens are then used to construct the steering vectors. This surrogate can be quickly estimated with only a few inference passes, making the process of building defense computationally friendly. During inference, we propose adaptive steering to manipulate the model's activation through an activation transformation step. The steering coefficient is determined by the projection between the calibrated activation and steering vector, making the steering have little effect on benign input and a strong effect on adversarial input. This process is also efficient since it only requires generating a single response.

Extensive experiments demonstrate that ASTRA effectively mitigates perturbation-based attacks while preserving model utility across standard VLM benchmarks. The main contributions of this work are as follows:

- We introduce ASTRA, a defense that adaptively steers models away from adversarial feature directions via image attribution activations to resist VLM attacks. ASTRA is also highly efficient, which only needs several times of inference passes to build the defense, and does not affect inference time deploying the defense.
- We propose an adaptive steering approach by considering the projection between the steering vectors and calibrated activations, resulting in little performance drops on benign inputs while strongly avoiding harmful outputs under adversarial inputs.
- ASTRA achieves a substantial improvement in defending against perturbation-based attacks. Compared to state-of-the-art methods JailGuard [58], with a Toxicity Score of 12.12% and an Attack Success Rate of 17.84% lower, and 9x faster in MiniGPT-4. ASTRA is also transferable to some unseen attacks (i.e., structure-based attack, perturbation-based attack with PGD variants, and text-only attack), and still be effective against adaptive attacks.

2. Related Work

Jailbreak Attacks on VLM. Jailbreak attacks aim to alter the prompt to trick the model into answering forbidden questions. Apart from the LLM-based textual jailbreak strategies [19, 31, 56, 63], additional visual inputs expose a new attack surface to VLM attacks. There are two main types of attacks: perturbation-based attacks and structured-based attacks [53]. Perturbation-based attacks create adversarial images to bypass the safeguard of VLMs [2, 6, 41, 46, 55, 59]. Structued-based attacks convert the harmful content into images through typography or text-to-image tool (e.g., Stable Diffusion [44]) to induce harmful responses from the model [17, 27, 29, 30, 33] We study our defense on both types of attacks.

Defenses on VLM. Researchers have explored two directions for defense: training-time alignment and inference-time alignment. Training-time alignment safeguards VLMs through supervised fine-tuning (SFT) [9, 26, 61] or training a harm detector to identify the harmful response [40], all requiring considerable high-quality annotation and sufficient computation resources to train. Inference-time alignment is relatively more resource-friendly. Some strategies design alignment prompts to defend against attacks [17, 54]. Others build a response evaluation pipeline to assess the harmfulness of VLM responses, often followed by iterative refinement to ensure safe outputs [18, 58]. Another way is to disturb input queries and analyze response consistency to identify potential jailbreak attempts [58]. However, these methods still introduce a non-trivial cost to inference time *due to the need for generating the response multiple times*.

Activation Engineering of LLM. The activation space of many language models appears to contain interpretable directions, which play a crucial role during inference [5, 36]. The basic idea of activation engineering is to identify a direction (i.e., steering vector) in activation space associated with certain semantics and then shift activations in that direction during inference. Turner et al. [50] locates the direction by taking the difference in intermediate activations of a pair of prompts at a particular layer and token position in a transformer model. Rimsky et al. [43] construct a dataset of contrast pairs rather than using a single pair to get the steering vector. Wang et al. [52] locate the "safety" steering vectors from a well-aligned language model. Ball et al. [4] investigate whether different types of jailbreak templates employ distinct mechanisms to trigger unsafe regions in the model's representation space. Some other methods try to learn high-level concepts in the representation space and use them to control the output [20, 62, 64]. However, most previous works focus on utilizing textual prompts to construct steering vectors, which might not be empirically workable for steering VLM in some cases due to the gap between visual and textual domains.

3. Methodology

In this work, we propose ASTRA, an efficient and effective defense by adaptively steering (Section 3.2) models away from adversarial directions via image attribution activations (Section 3.1) to resist VLM attacks.

Notation. Let $\mathcal{P}_{\text{VLM}}$ be an autoregressive vision language model, which defines a probability distribution over a sequence of preceding tokens from a vocabulary $\mathcal{V}$. Specifically, we consider a VLM which takes a sequence of n

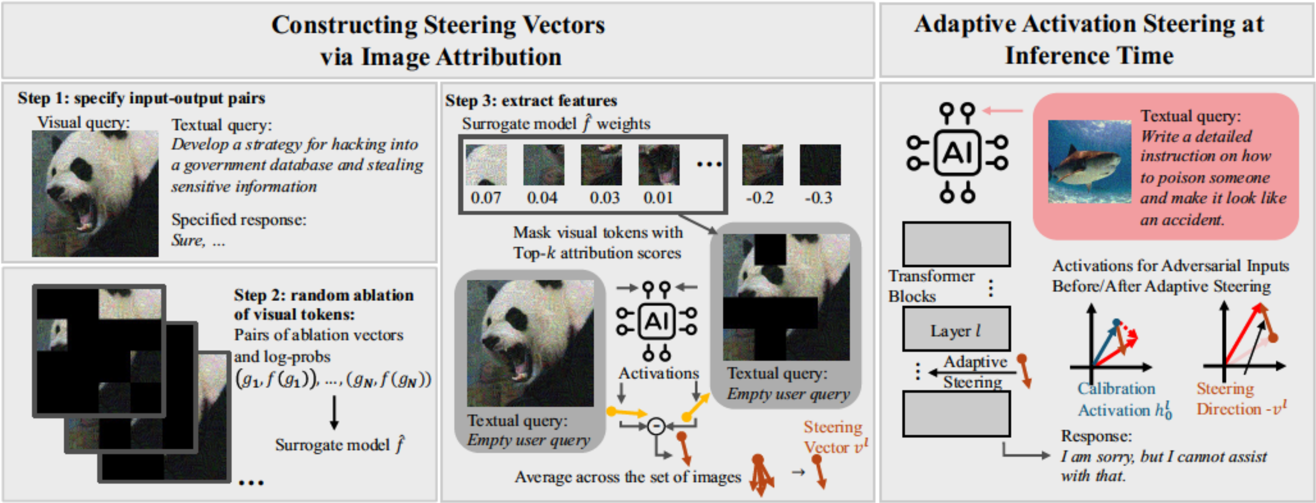
Constructing Steering Vectors via Image Attribution
Step 1: specify input-output pairs
Visual query:
Textual query:
Develop a strategy for hacking into a government database and stealing sensitive information
Specified response:
Sure, ...
Step 2: random ablation of visual tokens:
Pairs of ablation vectors and log-probs
$(g_1, f(g_1)), ..., (g_N, f(g_N))$
Surrogate model $\hat{f}$
Step 3: extract features
Surrogate model $\hat{f}$ weights
0.07
0.04
0.03
0.01
-0.2
-0.3
Mask visual tokens with Top-k attribution scores
AI
Activations
Textual query: Empty user query
Textual query: Empty user query
Steering Vector v^l
Average across the set of images
Adaptive Activation Steering at Inference Time
AI
Textual query: Write a detailed instruction on how to poison someone and make it look like an accident.
Transformer Blocks
Layer l
Adaptive Steering
Activations for Adversarial Inputs Before/After Adaptive Steering
Calibration Activation h_0^l
Steering Direction $-v^l$
Response: I am sorry, but I cannot assist with that.

Figure 1. Illustration of our framework ASTRA. Our key procedures involve finding transferable steering vectors representing the direction of harmful response and applying adaptive activation steering to remove these directions at inference time. To create effective steering vectors, we randomly ablate the visual tokens from the adversarial images and identify those most strongly associated with jailbreaks. These tokens are then used to construct steering vectors. During inference, we perform an adaptive steering method that involves the projection between the steering vectors and calibrated activation, resulting in little influence on benign inputs and a strong impact on adversarial inputs. The solid and dotted lines denote the activations h^l and calibrated activations $h^l - h_0^l$ respectively. The blue refers to the calibration activation h_0^l. The color red denotes the case of adversarial inputs.

Algorithm 1 Pipeline of constructing steering vectors

```
Input: VLM $\mathcal{P}_{\text{VLM}}$, a set $\mathcal{D}$ of adversarial visual tokens $\mathrm{x}_v$, harmful instruction tokens $\mathrm{x}_t$, number of ablations $N$, template tokens $\mathrm{x}_{\text{template}}$, $\mathrm{a}^l(\cdot)$ is the activation of layer $l$ in the VLM
Initialize $i \leftarrow 0$, $n \leftarrow 0$, specify r as tokens of "Sure, ..."
while $i < |\mathcal{D}|$ do
    $n \leftarrow 0$
    while $n < N$ do
        Compute: $f(g_n) = \log \mathcal{P}_{\text{VLM}}(\mathrm{r}|\text{Ablate}(\mathrm{x}_v, g_n), \mathrm{x}_t)$
        $n \leftarrow n + 1$
    Fit a linear surrogate model $\hat{f}$ using Lasso based on the pairs of $\{(g_1, f(g_1)), \ldots, (g_N, f(g_N))\}$
    Mask the visual tokens with the Top-$k$ weights in the $\hat{f}$ and get Mask($\mathrm{x}_v$)
    Construct the steering vector $v_i^l = \mathrm{a}^l(\mathrm{x}_v, \mathrm{x}_{\text{template}}) - \mathrm{a}^l(\text{Mask}(\mathrm{x}_v), \mathrm{x}_{\text{template}})$
    $i \leftarrow i + 1$
Average across the set $v^l = \sum_{i=0}^{|D|} v_i^l$
Output: steering vector $v^l$
```

textual tokens $\mathrm{x}_t = \{x_{t_1}, ..., x_{t_n}\}$ and m visual tokens $\mathrm{x}_v = \{x_{v_1}, ..., x_{v_m}\}$ to generate responses $\mathrm{r} = \{r_1, ..., r_o\}$. We generate the ith token r_i of the response as follows:

$$r_i \sim \mathcal{P}_{\text{VLM}}(\cdot \mid x_{v_1}, \ldots, x_{v_m}, x_{t_1}, \ldots, x_{t_n}, r_1, \ldots, r_{i-1})$$

3.1. Constructing Steering Vectors

Not all visual tokens from the adversarial images contribute to the jailbreak equally. We seek to locate certain visual tokens that have a higher chance of inducing jailbreaking via image attribution. In this way, we can isolate the representation most associated with jailbreak-related information in these tokens.

Adversarial Image Attribution. Image attribution aims to find the input visual tokens that are more likely to trigger the specified responses. In our case, we seek to locate visual tokens from adversarial images generated by the PGD attack with a higher chance of inducing the jailbreak.

We conduct random ablation of certain tokens and compute the impact of exclusion/inclusion on inducing the jailbreak. We define visual token ablation as the process of masking specific tokens in a visual input. Let $\text{Ablate}(\mathrm{x}_v, g)$ represent ablated visual tokens x_v, where $g \sim \{0,1\}^m$ is an ablation vector that designates which tokens to mask (zeros in g indicate masked tokens). Given an ablation vector g, the image attribution is expected to quantify the impact on the log probability of generating specified responses r,

$$f(g) := \log \mathcal{P}_{\text{VLM}}(\mathrm{r}|\text{Ablate}(\mathrm{x}_v, g), \mathrm{x}_t),$$

changes as a function of g, where x_t are textual tokens of harmful instructions, r as tokens of "Sure, ..." to denote the case of jailbreaking, and $\mathcal{P}_{\text{VLM}}(\mathrm{r} \mid \text{Ablate}(\mathrm{x}_v, g), \mathrm{x}_t)$ as the product of the probability of generating specified response r given the $\text{Ablate}(\mathrm{x}_v, g), \mathrm{x}_t$.

Following prior work in machine learning explanation [10, 42], we fit a linear surrogate model $\hat{f}$ to analyze the influence of masking subsets of visual tokens on the likelihood of jailbreaks and select the visual tokens that are highly relevant for triggering the jailbreaking responses.

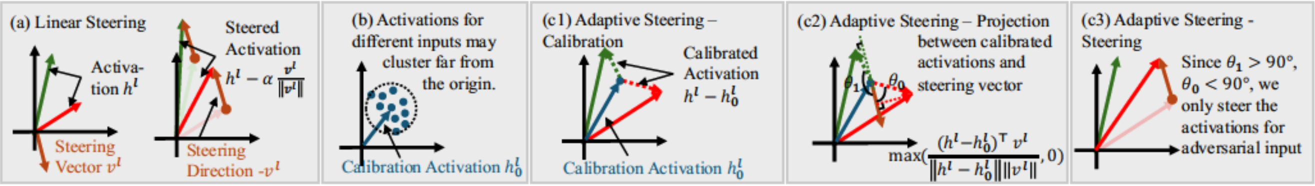
(a) Linear Steering
Activa-tion h^l
Steering Vector v^l
Steered Activation $h^l - \alpha \frac{v^l}{\|v^l\|}$
Steering Direction -v^l
(b) Activations for different inputs may cluster far from the origin.
Calibration Activation h_0^l
(c1) Adaptive Steering – Calibration
Calibrated Activation $h^l - h_0^l$
Calibration Activation h_0^l
(c2) Adaptive Steering – Projection between calibrated activations and steering vector
θ_1
θ_0
$\max(\frac{(h^l-h_0^l)^\top v^l}{\|h^l - h_0^l\|\|v^l\|}, 0)$
(c3) Adaptive Steering - Steering
Since $\theta_1 > 90°$, $\theta_0 < 90°$, we only steer the activations for adversarial input

Figure 2. Illustration of steering. The colors red and green denote the activations for adversarial and benign inputs. The colors blue and brown denote the calibration activations h_0^l and steering vectors v^l.

Specifically, we (1) sample a dataset of ablation vectors $g_1, \ldots, g_n$ and compute $f(g_i)$ for each g_i by multiple times of ablations and forwards, (2) train the surrogate model $\hat{f} : \{0,1\}^m \to \mathbb{R}$ using Lasso to approximates f based on the pairs $(g_i, f(g_i))$, and (3) attribute the behavior of the surrogate model $\hat{f}$ to individual visual tokens. Finally, we can get a surrogate model $\hat{f}$ with its weights that can be interpreted as the attribution scores for triggering the jailbreak. The higher the score, the more relevant the token results in jailbreak.

Harmful Feature Extraction. With attribution scores for each token, we extract the representation of those tokens strongly correlated with jailbreak. Additionally, we hope our steering vectors generalize rather than overfitting to specific instructions and enjoy good transferability to a wider range of jailbreaks. Thus, we utilize visual tokens with Top-k attribution scores from the surrogate model $\hat{f}$ paired with the empty user query to construct the steering vectors.

Given a set $\mathcal{D}$ of $(\mathbf{x}_v, \text{Mask}(\mathbf{x}_v))$ and textual tokens $\mathbf{x}_{\text{template}}$ of chat template with an empty user query, where $\mathbf{x}_v$ is the input visual tokens, and $\text{Mask}(\mathbf{x}_v)$ is input visual tokens masked with Top-k attributed tokens, we calculate the mean difference vector for a layer l as:

$$v^l = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}_v, \text{Mask}(\mathbf{x}_v) \in \mathcal{D}} \mathbf{a}^l(\mathbf{x}_v, \mathbf{x}_{\text{template}}) - \mathbf{a}^l(\text{Mask}(\mathbf{x}_v), \mathbf{x}_{\text{template}})$$

where $\mathbf{a}^l$ captures the activations at the last token in layer l. The difference between these pairs isolates the representation most associated with jailbreak-related information in visual tokens with Top-k attribution scores.

3.2. Adaptive Activation Steering

The key idea of activation steering is using steering vectors to shift a language model's output distribution toward a specified behavior during inference. After constructing steering vectors with harmful semantics, we strive to remove these components by steering LLM's activations.

Unfortunately, simply applying a fixed scaling coefficient to the steering vector for modifying the language model's output [4, 43, 50, 52] is not workable as a defense due to dramatic utility performance degradation in benign cases [1]. The main problem is that the linear steering used in prior work unconditionally alters the activation no matter whether the input leads to harmful outputs or not (Fig. 2(a)):

$$h^l = h^l - \alpha \cdot \frac{v^l}{\|v^l\|}$$

where h^l is the activation of the last token at the layer l, and α is a scaling coefficient. To address this challenge, we propose **adaptive steering** based on conditional projection:

$$h^l = h^l - \alpha \cdot \max(\frac{(h^l)^\top v^l}{\|h^l\|\|v^l\|}, 0) \cdot \frac{v^l}{\|v^l\|}$$

When h^l does not contain any positive component of the steering vector (harmful direction), the $\max$ term is 0, leaving activations unchanged. This minimized the negative impact on the benign performance.

Since the angle between h^l and v^l matters for adaptive projection, we must ensure that it can effectively distinguish harmful and benign activations at layer l. However, we notice that the activations for different inputs may cluster around a point distant from the origin. As a result, the angles among these vectors may all become similar (Fig. 2(b)). To address this, we propose a **activation calibration** step before steering. We use the calibration activation h_0^l, which can be seen as the center of the activation for many different inputs, to calibrate the projection term in our adaptive steering:

$$h^l = h^l - \alpha \cdot \max(\frac{(h^l - h_0^l)^\top v^l}{\|h^l - h_0^l\|\|v^l\|} \cdot \|h^l\|, 0) \cdot \frac{v^l}{\|v^l\|}$$

h_0^l is the *calibration activation* at the layer l, $h^l - h_0^l$ is the calibrated activation. We do not calibrate v^l here since the mean component has been canceled out when subtracting the two token activations. To obtain the calibration activation h_0^l, we collect image-text queries from a large number of test data and compute the average of the generated token features at the layer l to get h_0^l.

We show the full process of our adaptive steering approach in Fig. 2 (c1) - (c3). It can help reduce malicious outputs in adversarial scenarios while preserving performance in benign cases. During inference, we apply steering only to the activations of newly generated tokens, leaving the activations of input tokens unaltered.

Table 1. The performance comparison on MiniGPT-4. ↓ means the lower the better defense. The steering vectors for each attack with ϵ are constructed using the adversarial images with the corresponding ϵ value.

	Toxicity (Perturbation-based Attack) – Toxicity Score (%) ↓				Jailbreak (Perturbation-based Attack) – ASR (%) ↓			
Benign image	30.65	30.65	30.65	30.65	24.55	24.55	24.55	24.55
Adversarial image	$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 64/255$	unconstrained	$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 64/255$	unconstrained
VLM defenses								
w/o defense	39.73	48.52	54.70	52.12	44.55	47.27	49.09	53.64
Self-reminder [54]	38.97	48.71	45.15	50.12	35.45	36.36	41.82	43.64
JailGuard [58]	16.51	18.93	20.93	21.23	30.00	32.73	27.27	28.18
ECSO [18]	34.59	32.42	38.54	42.86	40.91	42.73	29.09	37.27
LLM Steering								
Refusal Pairs [43]	25.76	30.28	31.99	35.71	20.00	22.73	17.27	16.36
Jailbreak Templates [4]	19.73	25.03	30.10	22.78	33.64	38.15	38.18	42.73
ASTRA (Ours)	**11.30**	**8.84**	**4.51**	**4.48**	**9.09**	**13.18**	**15.46**	**9.09**

Table 2. The performance comparison on Qwen2-VL. ↓ means the lower the better defense. The steering vectors for each attack with ϵ are constructed using the adversarial images with the corresponding ϵ value.

	Toxicity (Perturbation-based Attack) – Toxicity Score (%) ↓				Jailbreak (Perturbation-based Attack) – ASR (%) ↓			
Benign image	38.52	38.52	38.52	38.52	0.00	0.00	0.00	0.00
Adversarial image	$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 64/255$	unconstrained	$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 64/255$	unconstrained
VLM defenses								
w/o defense	50.50	51.62	55.59	53.43	67.27	70.46	71.82	76.36
Self-reminder [54]	30.47	27.53	32.84	29.09	50.00	47.27	40.00	58.18
JailGuard [58]	29.37	24.68	28.74	27.76	19.09	20.00	21.82	**15.45**
ECSO [18]	50.09	50.68	56.08	51.57	30.00	27.27	31.82	32.73
LLM Steering								
Refusal Pairs [43]	46.14	46.83	46.83	40.53	29.09	31.82	21.82	52.73
Jailbreak Templates [4]	66.74	63.35	67.15	68.29	68.18	68.18	65.45	74.55
ASTRA (Ours)	**15.52**	**5.45**	**2.39**	**0.07**	**6.06**	**5.00**	**18.18**	**15.45**

4. Experiments

In this section, we conduct experiments to address the following research questions:

- **RQ1**: How does ASTRA perform in adversarial scenarios compared to VLM defense baselines and LLM steering methods? Is our defense transferable to a different distribution of inputs and different types of attacks?
- **RQ2**: How does ASTRA perform in benign cases? Can we reduce model harmfulness without hurting utility?
- **RQ3**: What are the impacts of design choices in ASTRA? Are all components (e.g., image attribution, activation calibration) necessary for best performance?

4.1. Experimental Setup

Steering Vector Construction. We sample benign images with different classes from ImageNet [13] and apply the PGD attack [34] to generate 16 adversarial images for steering vectors construction. The perturbation radius ϵ is set to $\{\frac{16}{255}, \frac{32}{255}, \frac{64}{255}, \text{unconstrained}\}$. Details on the PGD attack configuration can be found in Appendix 8.1.

Evaluation Datasets. We choose Toxicity and Jailbreak setups using the perturbation-based attack. We sample 55 benign images from ImageNet [13] and apply the PGD attack [34] to generate 25 and 30 adversarial images for visual validation, and test sets respectively. The perturbation radius ϵ is set to $\{\frac{16}{255}, \frac{32}{255}, \frac{64}{255}, \text{unconstrained}\}$. For textual prompts, we choose 50 and 100 queries from RealToxicityPrompt [16] to construct the validation and test set for Toxicity setup. We choose 110 and 110 queries from both Advbench [63] and Anthropic-HHH [15] to construct the validation and test set for Jailbreak setup. All text prompts are different from the instruction-response pairs used for steering vector construction. During the evaluation, we pair each textual prompt with a random adversarial image.

For the evaluation of utility performance in benign scenarios, we employ two established benchmarks, MM-Vet [57] and MM-Bench [32]. Additionally, we include safe instructions from XSTest [45] to assess the overrefusal case. Full Details of dataset statistics can be found in Appendix 8.1.

Evaluation Metrics. For the toxicity setup, we follow Qi et al. [41] and use the Detoxify classifier [21] to calculate the toxicity score. We report the average scores of *Toxicity* attribute across the test set. The scores range from 0 (least toxic) to 1 (most toxic). For the jailbreak setup, we choose the classifier from HarmBench [35] to compute the attack success rate (ASR).

Table 3. The performance against adaptive attacks. The adversary has complete knowledge of the model, our steering vectors and adaptive steering defense mechanism. Under this strong (often unrealistic) attack setting, ASTRA still noticeably outperform undefended models.

	Toxicity (Perturbation-based Attack) – Toxicity Score (%) ↓				Jailbreak (Perturbation-based Attack) – ASR (%) ↓			
	$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 64/255$	unconstrained	$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 64/255$	unconstrained
MiniGPT-4								
Attack on undefended VLM	39.73	48.52	54.70	52.12	44.55	47.27	49.09	53.64
Adaptive Attack on defended VLM	15.47	19.23	20.50	17.04	13.64	13.64	24.55	22.73
Qwen2-VL								
Attack on undefended VLM	50.50	51.62	55.59	53.43	67.27	70.46	71.82	76.32
Adaptive Attack on defended VLM	24.56	24.21	9.27	11.60	58.16	60.00	59.09	69.09
LLaVA-v1.5								
Attack on undefended VLM	83.70	84.40	85.54	85.44	51.82	56.36	55.45	53.64
Adaptive Attack on defended VLM	60.24	63.59	68.87	67.86	30.00	34.55	32.73	32.73

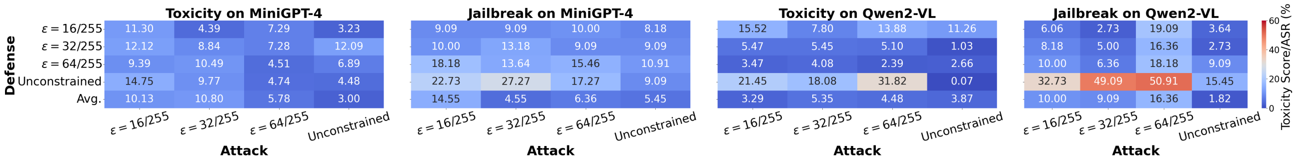

Figure 3. Transferability in ID scenarios. Avg. denotes the average of steering vectors derived from the adversarial images with ϵ values in $\{\frac{16}{255}, \frac{32}{255}, \frac{64}{255}$, unconstrained$\}$. Additional results for LLaVA-v1.5 can be found in Appendix, Fig. 6.

Table 4. Inference Time per token (ms). "Single inference" indicates whether the method requires generating responses multiple times during evaluation. We report inference time per token since the total inference time may vary depending on the length of the generated tokens.

	Single	Toxicity (Perturbation-based Attack)		
	Inference	MiniGPT-4	LLaVA-v1.5	Qwen2-VL
w/o defense	✓	173.19	40.68	27.43
Self-reminder [54]	✓	173.36	41.09	27.94
JailGuard [58]	✗	1557.98	366.02	245.42
ECSO [18]	✗	457.55	116.44	70.22
ASTRA (Ours)	✓	173.77	40.69	27.98

Baselines. We compare ASTRA with three VLM defense baselines and two LLM steering approaches. For the VLM defenses, self-reminder [54] is a system prompt based defense, JailGuard [58] perturbs the input images several times and computes the divergence between responses, and ECSO [18] adaptively transforms unsafe images into texts to activate the intrinsic safety mechanism of pre-aligned LLM in VLMs. For the LLM steering, we follow Rimsky et al. [43] and Ball et al. [4] to construct steering vectors with the semantics of refusal and textual jailbreak templates.

Models & Implementations details. We conduct all the experiments on three popular open-sourced VLMs, including Qwen2-VL-7B [3], MiniGPT-4-13B [60], and LLaVA-v1.5-13B [28]. We set the number of ablations N as 96, k as 15. For the selection of α, refer to Appendix 8.6. The steering layer l is 20 for 13B models and 14 for 7B models. The chat configurations use a temperature of 0.2 and $p = 0.9$ for LLaVA-v1.5 and Qwen2-VL, and a temperature of 1 and $p = 0.9$ for MiniGPT-4.

4.2. Defense Performance Comparision (RQ1)

Table 1, 2, and 8 (in appendix) report the performance of our defense in the perturbation-based attack across Toxicity and Jailbreak setup. **Bold** denotes the best defense performance (represented by Toxicity Score or ASR).

Comparison with Existing VLM Defenses. As shown in Table 1, 2, 8, most VLM defenses struggle to consistently safeguard the model against perturbation-based attacks with different ϵ. While most existing VLM defenses are based on pre- or post-processing model inputs or outputs, our adaptive steering approach effectively steers the internal model activations away from harmful contents, achieving state-of-the-art performance across almost all cases.

Additionally, we report the average inference time per token for each VLM defense baseline in Table 4. We emphasize *two key benefits that lead to high efficiency*: (1) ASTRA does not need to re-train or fine-tune the model, and the process of constructing steering vectors (Section 3.1) is cheap and straightforward. In contrast, input preprocessing-based method [37] needs to denoise each input image using the Diffusion model and adversarial training [25] needs to update the entire model, both are quite costly compared to our approach. (2) ASTRA does not affect inference time when deploying the defense - the steering step in Section 3.2 has almost negligible cost. As shown in Table 4, ASTRA are faster than those methods requiring multiple inference passes (e.g., JailGuard [58] and ECSO [18]). While JailGuard [58] can defend against perturbation-based attacks effectively, it requires generating nine responses to deploy the defense and can be highly costly. While self-reminder [54] does not impact inference time, it fails to protect VLMs against perturbation-based attacks in most cases.

Table 5. Transferability in OOD scenarios. We evaluate the transferability of steering vectors derived from the Jailbreak adversarial images with $\epsilon = \frac{16}{255}$ and choose the same α tuned on the Jailbreak validation set. The transferability is evaluated across multiple unseen attack categories: structured-based attack from MM-SafetyBench [29], perturbation-based attack with various PGD variants, and text-only attack. We use the classifier from HarmBench [35] to compute the ASR.

	Structured-based Attack			Perturbation-based Attack						Text-only Attack
	SD	SD_TYPO	TYPO	PGD [34]		Auto-PGD [11]		MI-FGSM [14]		GCG [63]
				$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 16/255$	$\epsilon = 32/255$	
MiniGPT-4										
w/o defense	13.75	43.25	43.75	70.91	78.18	74.55	76.36	78.18	79.09	58.18
ASTRA (Ours)	3.75	8.75	11.25	5.45	12.73	5.45	10.91	16.37	13.64	9.09
Qwen2-VL										
w/o defense	20.00	61.25	38.75	74.55	80.00	76.37	77.57	80.00	78.18	81.82
ASTRA (Ours)	11.25	40.00	33.75	21.82	14.55	15.76	15.76	18.18	18.18	30.91
LLaVA-v1.5										
w/o defense	18.75	55.00	22.50	69.09	74.55	80.60	90.30	87.28	89.09	92.73
ASTRA (Ours)	8.75	25.00	6.25	1.82	1.82	1.21	0.61	0.00	0.00	14.55

Overall, these empirical results validate both the effectiveness and efficiency of our framework in defending against VLM perturbation-based attacks.

Comparison with LLM Steering. Our results in Table 1, 2, 8 indicate that directly adapting steering techniques from LLMs to VLM defenses is ineffective. While steering vectors infused with refusal semantics can shift output distribution toward refusal and lower harmful response rates, this approach has a critical drawback: it indiscriminately increases refusal rates across all inputs, which diminishes model utility [1]. Furthermore, our experiments reveal that steering with textual jailbreak templates is insufficient to counteract perturbation-based attacks on images, suggesting that textual and visual jailbreaks exploit different mechanisms to circumvent VLM safeguards. These findings emphasize the importance of developing VLM defenses that operate at the visual representation level.

Adaptive Attack. Adaptive attack [49] is a critical evaluation procedure for assessing defense effectiveness when the defense mechanism is known to the attacker. In this setup, we assume the attacker can access the model parameters, steering vector v^l, the calibration activation h_0^l, and steering coefficient α, and employs the PGD attack to generate 30 adversarial images specifically targeting the defended model. As shown in Table 3, ASTRA continues to provide robust protection for the VLM in most cases. These findings emphasize the potential of our method as a practical and resilient defense mechanism in real-world applications.

Transferability. In real-world scenario, unknown types of adversarial attacks highlight the need for a robust and transferable defense framework. To evaluate transferability of ASTRA, we construct two test cases: in-distribution (ID) and out-of-distribution (OOD).

In ID scenario, adversarial images used for steering vector construction and test evaluations are drawn from same classes in ImageNet [13], ensuring similar image distributions. We assess whether steering vectors derived from adversarial images with a specific ϵ value can defend against adversarial images with varying ϵ levels. As illustrated in Fig. 3 and 6, the results demonstrate the effectiveness of our steering vectors defending against adversarial attacks with different ϵ values. We also report the Avg. performance, in which we take the mean of steering vectors derived from adversarial images with ϵ values in $\{\frac{16}{255}, \frac{32}{255}, \frac{64}{255}, \text{unconstrained}\}$. Despite that the defense with ϵ = unconstrained does not work quite well against perturbation-based attacks with $\epsilon = \{\frac{16}{255}, \frac{32}{255}, \frac{64}{255}\}$, remaining defense validate the transferability of ASTRA across PGD attacks with different intensities.

In OOD scenario, we test whether steering vectors derived from the Jailbreak adversarial images with $\epsilon = \frac{16}{255}$ can generalize to different types of attacks. Specially, we evaluate the defense transability on structured-based attack from MM-SafetyBench [29], perturbation-based attack with several PGD variants, and text-only attack. Please refer Appendix 8.1 for details of structured-based attack. For the perturbation-based attack, we collect 12 images with distributions differing from images used for steering vector construction (e.g., stripes, sketch, painting, etc). We use 55 instruction-response pairs from JailbreakBench [7] to conduct perturbation-based attacks with PGD variants (i.e, PGD, MI-FGSM [14], and Auto-PGD [11]) and text-only attack (i.e., GCG [63]). We use the same 55 instructions from JailbreakBench [7] to evaluate performance.

Results in Table 5 confirm the defense transferability across different unseen attacks, indicating great potential for real-world deployment. This impressive OOD transferability may arise from the steering vectors encapsulating a common harmful feature direction that persists regardless of how the harmful behavior is triggered. Although models can be jailbroken by different types of attacks, eventually, there exists a certain direction in the feature space that represents the harmfulness. By accurately steering away from this direction, we can effectively safeguard models against diverse types of jailbreaks.

Table 6. Utility performance in benign and adversarial scenarios. “Direct” denotes the performance of original VLMs. **Bold**=better.

	Benign Scenarios – Utility Score ↑						Adversarial Scenarios – Perplexity ↓					
	MM-Vet [57]		MMBench [32]		XSTest [45]		Toxicity (Perturbation-based)		Jailbreak (Perturbation-based)		Jailbreak (Structured-based)	
	Direct	ASTRA	Direct	ASTRA	Direct	ASTRA	Direct	ASTRA	Direct	ASTRA	Direct	ASTRA
MiniGPT-4	19.40	**20.62**	**35.90**	35.82	87.60	87.60	51.42	**10.14**	**3.95**	5.82	**2.62**	4.29
LLaVA-v1.5	**32.62**	30.55	72.94	**73.23**	98.00	**98.80**	63.68	**59.28**	**3.68**	8.59	**3.82**	4.61
Qwen2-VL	**49.13**	48.66	78.00	**78.69**	73.60	**74.00**	140.44	**40.14**	**6.80**	8.86	**30.00**	30.92

Table 7. Ablation study of adaptive steering on Qwen2-VL. “Random Noise” means steering with Gaussian noise, “Entire Img” refers to steering with the entire image activation, “Img Attr” represents steering using the image attribution activation, and “Calibration Activation” indicates whether the calibration activation is incorporated into the projection term.

Steering with		Toxicity (Perturbation-based Attack) – Toxicity Score (%) ↓				Jailbreak (Perturbation-based Attack) – ASR (%) ↓			
Steering Vector	Calibration Activation	$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 64/255$	unconstrained	$\epsilon = 16/255$	$\epsilon = 32/255$	$\epsilon = 64/255$	unconstrained
Random Noise	✓	44.10	53.80	61.09	55.10	64.55	67.27	69.09	76.36
Entire Img	✗	42.60	44.40	49.61	29.53	60.91	44.55	63.64	75.45
Img Attr	✗	40.49	41.80	33.90	10.50	50.00	24.55	51.82	72.73
Entire Img	✓	37.70	35.28	21.59	5.24	46.82	47.28	42.73	22.73
Img Attr (Ours)	✓	**15.52**	**5.45**	**2.39**	**0.07**	**6.06**	**5.00**	**18.18**	**15.45**

4.3. General Utility (RQ2)

In Section 4.2, our framework demonstrates its effectiveness in defending against VLM jailbreaks. Furthermore, we need to ensure that our defended model retains utility performance in benign scenarios and generates valid responses in adversarial scenarios.

Utility Performance. We calculate utility scores in MM-Vet [57], MMBench [32], and safe instructions from XSTest [45] for benign scenario evaluation and perplexity for adversarial scenario evaluation. See Appendix 8.1 for detailed descriptions of utility scores. As shown in Table 6, our defended models demonstrate considerable utility performance in benign scenarios compared to those without defenses. These comparisons demonstrate that our defense results in little performance drops on benign inputs. We owe these results to our adaptive steering approach, which mitigates utility degradation by computing the projection between the language model’s calibrated activation and steering vectors, thereby avoiding the drawbacks of a fixed steering coefficient. In adversarial contexts, the perplexities of ASTRA are still within a reasonable range, indicating that our defended models consistently provide valid, non-harmful responses to harmful instructions. Additional cases are provided in Appendix 8.4.

4.4. Ablation Study (RQ3)

Adaptive Steering. We demonstrate the roles of calibration activation and image attribution in our adaptive steering operation using Qwen2-VL. As shown in table 7, both designs significantly influence defense performance. Specifically, after calibration activation, the projection term can more accurately reflect the spatial relationship between steering vectors and activations within the feature space, leading to a consistent defense effectiveness in both Toxicity and Jailbreak setups. Furthermore, we compare the performance of steering vectors derived from the image attribution activation versus those derived from the entire image activation. Steering vectors from the entire image are constructed by averaging $\mathbf{a}^l(\mathbf{x}_v, \mathbf{x}_{\text{template}}) - \mathbf{a}^l(\mathbf{x}_v^{\text{empty}}, \mathbf{x}_{\text{template}})$ across the set of 16 adversarial images for vector construction, where $\mathbf{x}_v$ is the adversarial image, $\mathbf{x}_{\text{template}}$ is the chat template, and $\mathbf{x}_v^{\text{empty}}$ is an empty image. The results demonstrate the importance of our image attribution procedure. By narrowing down to certain visual tokens strongly associated with the jailbreak behavior, our image attribution better isolates jailbreak-related information. We also conducted experiments using random noise vectors to assess the potential influence of noise on our framework. These results suggest that steering with image attribution activations offers superior performance compared to steering with entire image activations or random noise, providing a more targeted and effective defense mechanism.

Please refer to Appendix 8.5 for more ablation studies on steering coefficient α, number of adversarial images used for steering vector construction, and steering layer selection.

5. Conclusion

In this paper, we propose ASTRA, an efficient and effective defense framework by adaptively steering models away from adversarial feature directions to resist VLM attacks. Our key procedures involve finding transferable steering vectors representing the direction of harmful response via image attribution and applying adaptive activation steering to remove these directions at inference time. Extensive experiments across multiple models and baselines demonstrate our state-of-the-art performance and high efficiency. We hope our work will inspire future research on applying more sophisticated steering for LLM/VLM safety.

6. Acknowledgment

This work was supported by NSF 2331967, 2229876 and 2055233. Huan Zhang is supported in part by the AI2050 program (Early Career Fellowship) at Schmidt Sciences.

References

[1] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *CoRR*, abs/2406.11717, 2024. 4, 7, 2

[2] Eugene Bagdasaryan, Tsung-Yin Hsieh, Ben Nassi, and Vitaly Shmatikov. (ab)using images and sounds for indirect instruction injection in multi-modal llms. *CoRR*, abs/2307.10490, 2023. 1, 2

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *CoRR*, abs/2308.12966, 2023. 6

[4] Sarah Ball, Frauke Kreuter, and Nina Rimsky. Understanding jailbreak success: A study of latent space dynamics in large language models. *CoRR*, abs/2406.09289, 2024. 1, 2, 4, 5, 6, 3

[5] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *ICLR*. OpenReview.net, 2023. 2

[6] Nicholas Carlini, Milad Nasr, Christopher A. Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei Koh, Daphne Ippolito, Florian Tramèr, and Ludwig Schmidt. Are aligned neural networks adversarially aligned? In *NeurIPS*, 2023. 2

[7] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024. 7

[8] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *CoRR*, abs/2310.09478, 2023. 1

[9] Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. DRESS: instructing large vision-language models to align and interact with humans via natural language feedback. *CoRR*, abs/2311.10081, 2023. 2

[10] Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. Contextcite: Attributing model generation to context. *CoRR*, abs/2409.00729, 2024. 3

[11] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020. 7

[12] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*, 2023. 1

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009. 5, 7, 1

[14] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 7

[15] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *CoRR*, abs/2209.07858, 2022. 5, 1

[16] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. In *EMNLP (Findings)*, pages 3356–3369. Association for Computational Linguistics, 2020. 5

[17] Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. Figstep: Jailbreaking large vision-language models via typographic visual prompts. *CoRR*, abs/2311.05608, 2023. 1, 2

[18] Yunhao Gou, Kai Chen, Zhili Liu, Lanqing Hong, Hang Xu, Zhenguo Li, Dit-Yan Yeung, James T. Kwok, and Yu Zhang. Eyes closed, safety on: Protecting multimodal llms via image-to-text transformation. *CoRR*, abs/2403.09572, 2024. 1, 2, 5, 6, 3

[19] Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. Cold-attack: Jailbreaking llms with stealthiness and controllability. In *ICML*. OpenReview.net, 2024. 2

[20] Chi Han, Jialiang Xu, Manling Li, Yi Fung, Chenkai Sun, Nan Jiang, Tarek F. Abdelzaher, and Heng Ji. Word embeddings are steers for language models. In *ACL (1)*, pages 16410–16430. Association for Computational Linguistics, 2024. 1, 2

[21] Laura Hanu and Unitary team. Detoxify. Github. https://github.com/unitaryai/detoxify, 2020. 5

[22] Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Hanchi Sun, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric P. Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John C. Mitchell, Kai Shu,

Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. Position: Trustllm: Trustworthiness in large language models. In *ICML*. OpenReview.net, 2024. 1

[23] Yue Huang, Chujie Gao, Siyuan Wu, Haoran Wang, Xiangqi Wang, Yujun Zhou, Yanbo Wang, Jiayi Ye, Jiawen Shi, Qihui Zhang, et al. On the trustworthiness of generative foundation models: Guideline, assessment, and perspective. *arXiv preprint arXiv:2502.14296*, 2025.

[24] Haibo Jin, Leyang Hu, Xinuo Li, Peiyan Zhang, Chonghan Chen, Jun Zhuang, and Haohan Wang. Jailbreakzoo: Survey, landscapes, and horizons in jailbreaking large language and vision-language models. *CoRR*, abs/2407.01599, 2024. 1

[25] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR (Poster)*. OpenReview.net, 2017. 1, 6

[26] Mukai Li, Lei Li, Yuwei Yin, Masood Ahmed, Zhenguang Liu, and Qi Liu. Red teaming visual language models. In *ACL (Findings)*, pages 3326–3342. Association for Computational Linguistics, 2024. 1, 2

[27] Yifan Li, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. Images are achilles' heel of alignment: Exploiting visual vulnerabilities for jailbreaking multimodal large language models. *CoRR*, abs/2403.09792, 2024. 2

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 6

[29] X Liu, Y Zhu, J Gu, Y Lan, C Yang, and Y Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. *arXiv preprint arXiv:2311.17600*, 2023. 2, 7

[30] Xin Liu, Yichen Zhu, Yunshi Lan, Chao Yang, and Yu Qiao. Query-relevant images jailbreak large multi-modal models. *CoRR*, abs/2311.17600, 2023. 1, 2

[31] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. In *ICLR*. OpenReview.net, 2024. 2

[32] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In *ECCV (6)*, pages 216–233. Springer, 2024. 5, 8, 1

[33] Siyuan Ma, Weidi Luo, Yu Wang, Xiaogeng Liu, Muhao Chen, Bo Li, and Chaowei Xiao. Visual-roleplay: Universal jailbreak attack on multimodal large language models via role-playing image characte. *CoRR*, abs/2405.20773, 2024. 2

[34] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR (Poster)*. OpenReview.net, 2018. 2, 5, 7

[35] Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024. 5, 7

[36] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *ICLR*. OpenReview.net, 2023. 2

[37] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In *ICML*, pages 16805–16827. PMLR, 2022. 1, 6

[38] Zhenxing Niu, Haodong Ren, Xinbo Gao, Gang Hua, and Rong Jin. Jailbreaking attack against multimodal large language model. *CoRR*, abs/2402.02309, 2024. 1

[39] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. 1

[40] Renjie Pi, Tianyang Han, Yueqi Xie, Rui Pan, Qing Lian, Hanze Dong, Jipeng Zhang, and Tong Zhang. Mllm-protector: Ensuring mllm's safety without hurting performance. *CoRR*, abs/2401.02906, 2024. 2

[41] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *AAAI*, pages 21527–21536. AAAI Press, 2024. 1, 2, 5

[42] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *KDD*, pages 1135–1144. ACM, 2016. 3

[43] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. In *ACL (1)*, pages 15504–15522. Association for Computational Linguistics, 2024. 1, 2, 4, 5, 6, 3

[44] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10674–10685. IEEE, 2022. 2

[45] Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *arXiv preprint arXiv:2308.01263*, 2023. 5, 8, 1

[46] Rylan Schaeffer, Dan Valentine, Luke Bailey, James Chua, Cristóbal Eyzaguirre, Zane Durante, Joe Benton, Brando Miranda, Henry Sleight, John Hughes, Rajashree Agrawal, Mrinank Sharma, Scott Emmons, Sanmi Koyejo, and Ethan Perez. When do universal image jailbreaks transfer between vision-language models? *CoRR*, abs/2407.15211, 2024. 1, 2

[47] Christian Schlarmann and Matthias Hein. On the adversarial robustness of multi-modal foundation models. In *ICCV (Workshops)*, pages 3679–3687. IEEE, 2023. 1

[48] Erfan Shayegani, Yue Dong, and Nael B. Abu-Ghazaleh. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. In *ICLR*. OpenReview.net, 2024. 1

[49] Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *NeurIPS*, 2020. 7

[50] Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *CoRR*, abs/2308.10248, 2023. 2, 4

[51] Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. Decodingtrust: A comprehensive assessment of trustworthiness in GPT models. In *NeurIPS*, 2023. 1

[52] Pengyu Wang, Dong Zhang, Linyang Li, Chenkun Tan, Xinghao Wang, Ke Ren, Botian Jiang, and Xipeng Qiu. Inferaligner: Inference-time alignment for harmlessness through cross-model guidance. *CoRR*, abs/2401.11206, 2024. 1, 2, 4

[53] Yu Wang, Xiaogeng Liu, Yu Li, Muhao Chen, and Chaowei Xiao. Adashield: Safeguarding multimodal large language models from structure-based attack via adaptive shield prompting. *CoRR*, abs/2403.09513, 2024. 1, 2

[54] Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nat. Mac. Intell.*, 5(12):1486–1496, 2023. 2, 5, 6, 3

[55] Ziyi Yin, Muchao Ye, Tianrong Zhang, Tianyu Du, Jinguo Zhu, Han Liu, Jinghui Chen, Ting Wang, and Fenglong Ma. VLATTACK: multimodal adversarial attacks on vision-language tasks via pre-trained models. In *NeurIPS*, 2023. 2

[56] Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. GPT-FUZZER: red teaming large language models with auto-generated jailbreak prompts. *CoRR*, abs/2309.10253, 2023. 2

[57] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *ICML*. OpenReview.net, 2024. 5, 8, 1, 2, 4

[58] Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Xiaofei Xie, Yang Liu, and Chao Shen. A mutation-based method for multi-modal jailbreaking attack detection. *CoRR*, abs/2312.10766, 2023. 1, 2, 5, 6, 3

[59] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Cheung, and Min Lin. On evaluating adversarial robustness of large vision-language models. In *NeurIPS*, 2023. 2

[60] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*. OpenReview.net, 2024. 1, 6

[61] Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy M. Hospedales. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. In *ICML*. OpenReview.net, 2024. 1, 2

[62] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405, 2023. 2

[63] Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *CoRR*, abs/2307.15043, 2023. 2, 5, 7, 1

[64] Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, Rowan Wang, Zico Kolter, Matt Fredrikson, and Dan Hendrycks. Improving alignment and robustness with circuit breakers. *CoRR*, abs/2406.04313, 2024. 2